





DUDLEY KNOX LIBRARY  
NAVAL POSTGRADUATE SCHOOL  
MONTREY, CALIFORNIA 93943-5000





# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



# THESIS

A R E S  
A SYSTEM FOR REAL-TIME  
OPERATIONAL AND TACTICAL DECISION SUPPORT

by

Antonios L. Vassiliou

December 1986

Thesis Advisor:

Gerald G. Brown

Approved for public release; distribution is unlimited.

T233692



## REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b RESTRICTIVE MARKINGS		
2a SECURITY CLASSIFICATION AUTHORITY			3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited		
2b DECLASSIFICATION/DOWNGRADING SCHEDULE					
4 PERFORMING ORGANIZATION REPORT NUMBER(S)			5 MONITORING ORGANIZATION REPORT NUMBER(S)		
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School		6b OFFICE SYMBOL (If applicable) 54		7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943 - 5000			7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943 - 5000		
8a NAME OF FUNDING/SPONSORING ORGANIZATION		8b OFFICE SYMBOL (If applicable)		9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c ADDRESS (City, State, and ZIP Code)			10 SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO	PROJECT NO	TASK NO
			WORK UNIT ACCESSION NO		
11 TITLE (Include Security Classification) ARES A SYSTEM FOR REAL-TIME OPERATIONAL AND TACTICAL DECISION SUPPORT					
12 PERSONAL AUTHOR(S) Vassiliou Antonios L.					
13a TYPE OF REPORT Master's Thesis		13b TIME COVERED FROM _____ TO _____		14 DATE OF REPORT (Year, Month, Day) 1986 December	
15 PAGE COUNT 44					
16 SUPPLEMENTARY NOTATION					
17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Decision Support System, Logistics Model,		
			Operational Assignment, Tactical Allocation,		
			Integral Programming, Generalized Networks.		
9 ABSTRACT (Continue on reverse if necessary and identify by block number)					
<p>We introduce a real - time decision support system which uses optimization methods, simulation, and judgement of the decision maker for <u>operational assignment</u> of military units to tasks and for <u>tactical allocation</u> of unit resources to task requirements.</p> <p>The system, named ARES for the Greek god of war, accommodates a high degree of detail in the logistics of unit movements and operations, yet separates the assignment and allocation activities in a fashion which naturally accommodates human intervention and judgement.</p> <p>ARES is designed to assist the decision maker, not to replace him. ARES is demonstrated with a hypothetical scenario constructed for Engineering Battalions of the Hellenic</p>					
0 DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS			21 ABSTRACT SECURITY CLASSIFICATION Unclassified		
2a NAME OF RESPONSIBLE INDIVIDUAL Gerald G. Brown			22b TELEPHONE (Include Area Code) 408-646-2140		22c OFFICE SYMBOL 55BW

## 19. ABSTRACT (Continued)

Army which are to be assigned 20 tasks employing 25 resource types in repairing major damage to public works following a large earthquake\* ARES is designed for use in real time, and quick data preparation is aided by the provision of standard task icons from published sources.

---

\* This hypothetical data was prepared prior to the earthquake in Kalamata near Athens on 13 September, 1986, and exhibits uncanny, but coincidental resemblance to that real situation.



Approved for public release, distribution is unlimited

A R E S  
A System for Real-Time  
Operational and Tactical Decision Support

by  
Antonios L. Vassiliou  
Major, Hellenic Army  
B.S., Military Academy of Greece, 1974

Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL  
December 1986

---

## ABSTRACT

We introduce a real-time decision support system which uses optimization methods, simulation, and judgement of the decision maker for operational assignment of military units to tasks and for tactical allocation of unit resources to task requirements.

The system, named ARES for the Greek god of war, accommodates a high degree of detail in the logistics of unit movements during operations, yet separates the assignment and allocation activities in a fashion which naturally accommodates human intervention and judgement.

ARES is designed to assist the decision maker, not to replace him. ARES is demonstrated with a hypothetical scenario constructed for 14 Engineering Battalions of the Hellenic Army which are assigned 20 tasks employing 25 resource types in repairing major damage to public works following a great earthquake\*.

ARES is designed for use in real time, and quick data preparation is aided by the provision of standard task icons from published sources.

---

\* This hypothetical data was prepared prior to the earthquake in Kalamata near Athens on 13 September, 1986, and exhibits uncanny, but coincidental resemblance to that real situation.

## TABLE OF CONTENTS

I.	INTRODUCTION.....	7
II.	OPERATIONAL ASSIGNMENT MODEL (IP).....	12
III.	OPERATIONAL ASSIGNMENT MODEL ( $IP_L$ ).....	15
IV.	TACTICAL ALLOCATION MODEL ( $GN_K$ ).....	18
V.	CONSIDERATION OF LOGISTICS.....	21
VI.	AN EXAMPLE SCENARIO.....	23
VII.	DESIGN AND IMPLEMENTATION.....	28
VIII.	SCENARIO RESULTS.....	32
IX.	COMPUTATIONAL EXPERIENCE.....	35
X.	DISCUSSION AND CO CLUSION.....	38
	LIST OF REFERENCES.....	41
	INITIAL DISTRIBUTION LIST.....	43



## ACKNOWLEDGEMENT

First of all I would like to express my most sincere appreciation and thanks to Professor Gerald G. Brown, my thesis advisor, with whose support, professional assistance, and indefatigable contribution and advise, this 'ARES' system has been successfully completed.

Also I would like to thank professor Benjamin J. Roberts, the second reader, for his suggestions and comments.

I am deeply indebted to my wife, Paraskeyi, and two sons, Constantine and Alexander, for their patience, understanding, and assistance.

Lastly I express my love to my newborn daughter Veronica whose cries inspired me to work harder during the long nights devoted to the study of this "ARES" system.

## I. INTRODUCTION

We introduce ARES, a prototypic system for real-time operational and tactical decision support. ARES is designed to quickly and effectively help respond to complex emergent problems in disaster relief, in the operational art and tactics of warfare, and in related multiperiod, large-scale employment of heterogeneous, substitutable resources restricted in availability and demand over time, over geography, and by organizational limitations.

Although a great deal of work has been done in strategic modelling in many contexts, there is relatively little available modelling help beyond simple thumb rules for the time-pressed (operational or tactical) decision maker to translate strategic goals into logistically constrained operational and tactical plans (and the issues are different). The luxuries of hypothetical additional resources and the time to analyze their employment are just not available in the operational and tactical domains: operational and tactical decisions must be made quickly, and usually involve employing only resources actually available to perform whatever mission is at hand.

The history of assignment and allocation models for planning emergency logistics extends back to some of the earliest work in linear programming, game theory, and their economic interpretation. We cite only a few of the references in this large body of literature. The seminal works by Dantzig and by Koopmans (both found in Koopmans 1951) are explicitly motivated by large-scale logistics problems. Karchere and Hoerber (1953) give early direction on the use of newly developed optimization technology in weapon

system planning and allocation, discussing substitutability of resources and choice of suitable objective functions. Geisler (1959) reports RAND's first use of man-machine simulation of logistics support activities. Chaiken and Larson (1972) state some basic issues in logistic location and task assignment for emergency service vehicles: how many units should there be, where should they be located, who should they serve, and how can they be relocated to substitute for units not available. Kaplan (1973) redeploys divisible resources with linear programming. Fitzsimmons (1973) states a nonlinear response time model and uses pattern search to locate units well and allocate workload equitably. Swoveland, Uyeno, Vertinsky, and Vickson (1973) employ simulation and human interaction to set up a unit location problem as a quadratic assignment model which is solved with an elegant heuristic. Bracken and McGill (1974) formulate strategic force planning models as two-sided games solved with nonlinear programming. Bracken, Falk, and Karr (1975) apply multiperiod, two-person zero-sum games formulated to develop strategies for unit sortie allocations. Finally, Kolesar and Walker (1974) develop a multi-stage solution approach to unit and task assignment using set covering and transportation-like integer linear programs which are used in real time by applying heuristics.

Named for the Greek god of war, ARES is a proof prototype of a real-time decision support system. It employs optimization and simulation to capture and exploit a high degree of realism without demanding unreasonable amounts of data, or locking the decision maker out of the decision process. The intent is to provide quick credible advice with good global perspective at a cost no greater than the relatively myopic



decision methods now widely used.

ARES accommodates enough detail to support realistic decisions, but not so much as to render the process useless. For the intended applications, the particular missions to be performed will not likely be known much in advance, but the generic types of missions are known and can be planned. ARES uses a taxonomy of prepared standardized icons for data describing possible missions. The idea is to help the decision maker quickly assemble a data scenario closely resembling the proximate situation from a menu of these standard icons.

We characterize the mission at hand as a set of geographically dispersed tasks, each composed of partially-ordered sub-tasks requiring over time varying amounts of different resources. Organizational units, also geographically dispersed and each possessing a different endowment of resources, are to be assigned responsibility for the tasks. Responsibility for each task rests with only one unit at any given time.

ARES consists of several models coordinated by a time interval simulator which also scales and manipulates scenario data in a fashion transparent to the decision maker. Two integer linear programs, a linear program, a georeference system, a mobility system, a decision-maker simulator, and extensive user interface and user override and control facilities complete the program suite. The models in ARES all use a standard data interface visible to the decision maker; this invites expansion with new models and features.

A scenario is created from the attributes of available units and task attributes derived in large part from standard cataloged data icons for similar tasks. A georeference system is accommodated to generate distance

costs and delays in relocating and operating units. The decision maker may preview the scenario and modify data or manually pre-assign tasks and units as he sees fit.

Operational assignment of tasks to units uses one of two integer programming models ( $IP_L$ ) or (IP). Good task aggregations for the unit assigned reduce unit relocation costs and match unit resource endowments with aggregated task resource requirements. Logistical considerations are paramount at this stage.

The decision maker can review the operational assignments, modify them manually, or reject them outright and restate the conditions for the original operational assignment scenario. An acceptable set of operational assignments is passed forward to a tactical model.

Tactical allocation of the resources of each unit to the requirements of its assigned tasks uses a linear programming model (GN). Substitutions among resources are permitted, although at reduced efficiencies in completing the tasks. Allocations recognize task priorities and the logistical effects of geographic proximity. In addition, unit efficiency in performing a particular task improves over time, and the sequence within tasks of resource requirements is considered. The result is a complete plan for each unit, showing what resources are to be used to fulfill each task requirement, and the efficiency with which operations are expected to be carried out. The allocation also determines which requirements will not be met in situations which overtax units.

Finally, the decision maker is presented with a complete solution, which he can accept, or modify, or reject outright and repeat. Regardless of his action, ARES is designed to lend quick insight. The decision maker

uses his own judgement concerning non-quantified factors, and he should gain a deeper understanding of the situation at hand from ARES.



## II. OPERATIONAL ASSIGNMENT MODEL (IP)

This integer program finds good aggregate assignments of tasks to units without explicit consideration of unit movement.

### Index Use

i	Tasks
j	Resources
k	Units

### Given Data

$d_{ik}$	Distance cost from unit k to task i
$\underline{r}_{ij}, \bar{r}_{ij}$	<u>Minimum</u> , <u>maximum</u> resource j requirements of task i
$\underline{a}_{jk}, \bar{a}_{jk}$	<u>Minimum</u> , <u>maximum</u> resource j employable by unit k
$\underline{z}_{jk}, \bar{z}_{jk}$	Penalties for violating <u>minimum</u> , <u>maximum</u> resource limits
$p_i$	Priority of task i (>0)
$\underline{u}_i, \bar{u}_i$	Penalties for not assigning, double-assigning task i
$f_{jj}$	Substitution efficiency of resource j (>0)
$s_{ij}$	Sequence of resource j requirement within task i
$h_{ijk}$	Consumption by task i of resource j from unit k

### Decision Variables

$x_{ik}$	Binary variable for assigning task i to unit k
----------	--

### Formulation

$$\begin{aligned}
 & \text{MIN } \sum_{ik} d_{ik} x_{ik} \\
 & \text{s. t. } \sum_k x_{ik} \stackrel{o}{=} (1,1) \quad ; (\underline{u}_i, \bar{u}_i) \quad \text{for all } i \quad (1) \text{ (GUB)} \\
 & \quad \sum_i h_{ijk} x_{ik} \stackrel{o}{=} (\underline{a}_{jk}, \bar{a}_{jk}) \quad ; (\underline{z}_{jk}, \bar{z}_{jk}) \quad \text{for all } j,k \quad (2) \\
 & \quad x_{ik} = \{0,1\} \quad \text{for all } i,k \quad (3) \quad \quad \quad (\text{IP})
 \end{aligned}$$

The notation  $\overset{\circ}{=} (\underline{r}, \overline{r}); (\underline{z}, \overline{z})$  indicates lower and upper ranges  $(\underline{r}, \overline{r})$  on row functional values with corresponding respective linear penalties per unit of violation  $(\underline{z}, \overline{z})$ ; i.e., this is a goal program with linear penalties, an elastic integer program (Brown and Graves 1975). Constraints (1) encourage assignment of each task to some unit, and form a generalized upper bound (GUB) row set (Dantzig and Van Slyke 1967). Constraints (2) express the goodness of fit of task assignments with employable unit resources. Constraints (3) preclude fractional assignment of tasks to units.

Consumption by task  $i$  of resource  $j$  from unit  $k$  is defined:

$$h_{ijk} \equiv \underline{r}_{ij} e^{-\ln f_{jj} + (p_i - 1)/10 + d_{ik}/\sigma_k + t_{ik}^{-2}}, \quad (1)$$

where  $\sigma_k$  is the speed of advance of a unit and  $t_{ik}$  is the number of periods that unit  $k$  has already been assigned task  $i$ . The rationale for the particular consumption function (1) amplifies the resource requirement  $\underline{r}_{ij}$  to account for the state of resource readiness  $f_{jj}$ , the task priority  $p_i$  (making less important tasks appear more expensive), the logistic proximity of unit  $k$  and task  $i$ ,  $d_{ik}/\sigma_k$ , and learning curve effect as a function of time since assignment,  $t_{ik}$ . The data are scaled so that (1) is in conformity with policy guidance or the judgement of the decision maker. Alternate consumption functions may appeal in other situations.

The distance costs  $d_{ik}$  and penalties  $\underline{u}_i$ ,  $\overline{u}_i$  and  $\underline{z}_{jk}$ ,  $\overline{z}_{jk}$  are expressed in commensurate units and deserve some thought by the modeller. For instance,  $\overline{z}_{jk}$  may be interpreted as how much additional distance cost should be incurred before considering overtaxing maximum resource

employment  $\bar{a}_{jk}$  for unit  $k$ ; this is a direct expression of logistical efficiency. For simplicity in our tests, distance costs  $d_{ik}$  are scaled by a policy parameter,  $\underline{z}_{jk}$  and  $\bar{z}_{jk}$  are part of the input script,  $\underline{u}_i$  is defined as  $100/p_i$ , and  $\bar{u}_i$  equals 100.



### III. OPERATIONAL ASSIGNMENT MODEL ( $IP_L$ )

The purpose of this integer program is to find good movements of units to locations from which they will be assigned good aggregate groups of tasks to perform.

#### Index Use

i	Tasks
j	Resources
k	Units
l	Locations (assumed here to be collocated with tasks)

#### Given Data

$d_{lk}$	Distance cost from unit k to location l
$g_{il}$	Distance cost from task i to location l
$\tilde{r}_{jli}$	Gross resource requirement j of task i performed from location l
$\tilde{a}_{jlk}$	Net resource availability j of unit k located at l

#### Decision Variables

$z_{il}$	Binary variable for assigning task i to location l
$x_{lk}$	Binary variable for moving unit k to location l

## Formulation

$$\begin{aligned} \text{MIN } \sum_{i,l} g_{il} z_{il} + \sum_{k,l} d_{lk} x_{lk} \\ \sum_l z_{il} &= (1,1); (\underline{u}_i, \bar{u}_i) \quad \text{for all } i \quad (1) \text{ (GUB)} \end{aligned}$$

$$\sum_l x_{lk} = (1,1); (m,m) \quad \text{for all } k \quad (2) \text{ (GUB)}$$

$$\sum_k x_{lk} = (0,1); (m,m) \quad \text{for all } l \quad (3)$$

$$-z_{il} + \sum_k x_{lk} = (0,1); (m,m) \quad \text{for all } i,l \quad (4)$$

$$-\sum_i \tilde{r}_{jil} z_{il} + \sum_k \tilde{a}_{jlk} x_{lk} = (0,0); (\underline{b}, \bar{b}) \quad \text{for all } l,j \quad (5)$$

$$z_{il} = \{0,1\} \quad \text{for all } i,l \quad (6)$$

$$x_{lk} = \{0,1\} \quad \text{for all } l,k \quad (7) \quad (IP_L)$$

(IP<sub>L</sub>) uses the notation of (IP). Constraints (1) encourage assignment of each task to some location. Constraints (2) allow movement of each unit to some location. (A GUB row set is formed by constraints (1) and (2).) Constraints (3) attempt to restrict assignments so at most one unit is moved to any particular location, (4) require that a unit be moved to any location to which a task is assigned, and (5) attempt to match for each location and each resource an aggregate assignment of tasks which have gross resource requirements about equal to the net resource availability of the unit moved to that location to perform the tasks. Constraints (6) and (7) preclude fractional location of tasks and units.

Gross resource requirement  $\tilde{r}_{jil}$  represents the resource  $j$  estimated to be required at location  $l$  in order that task  $i$  actually receive  $\underline{r}_{ij}$ .

$$\tilde{r}_{jil} \equiv \underline{r}_{ij} e^{-\ln f_{jj} + (p_i - 1)/10 + g_{il}/\sigma}, \quad (2)$$

where  $\sigma$  expresses the logistic radius of influence from any location; we

have used  $\sigma = 100$ . The gross resource requirement (2) amplifies the resource requirement  $r_{ij}$  in the same fashion as (1).

Net resource availability  $\tilde{a}_{jlk}$  represents the amount of resource  $j$  which unit  $k$  can deliver from its endowment  $\bar{a}_{jk}$  forward to location  $l$ . Unit  $k$  may be moving toward location  $l$  while supplying this net resource.

$$\tilde{a}_{jlk} \equiv \bar{a}_{jk} f_{jj} (\alpha_{lk} + (1-\alpha_{lk}) e^{-d_{lk}/\sigma_k}), \quad (3)$$

where  $\alpha_{lk}$  is the fraction of time which the unit will spend at its destination location, and  $\sigma_k$  is the speed of advance.

The distance costs  $d_{lk}$  and  $g_{il}$ , and the penalties  $\underline{u}_i$ ,  $\bar{u}_i$ ,  $m$ ,  $\underline{b}$  and  $\bar{b}$  all render the same objective function units. In our work,  $m \equiv 100$ , and  $\underline{u}_i$  and  $\bar{u}_i$  are defined as in (IP). The penalties for assigning too little (or too much) resource  $j$  to location  $l$  are  $\underline{b}$  (or  $\bar{b}$ ). We have used  $\underline{b} = 0.1$  and  $\bar{b} = 0.01$ .

#### IV. TACTICAL ALLOCATION MODEL ( $GN_k$ )

This linear program allocates resources to the tasks assigned to unit k.

##### Index Use

i	Tasks
j	Resources
w	Work (resources required by tasks assigned to unit k)

##### Given Data

$\underline{r}_{iw}, \bar{r}_{iw}$	<u>Minimum</u> , <u>maximum</u> work requirements w of assigned task i
$\underline{q}_{iw}, \bar{q}_{iw}$	Penalties for violating <u>minimum</u> , <u>maximum</u> work requirements
$\underline{a}_{jk}, \bar{a}_{jk}$	<u>Minimum</u> , <u>maximum</u> resource j employable by unit k
$\underline{z}_{jk}, \bar{z}_{jk}$	Penalties for violating <u>minimum</u> , <u>maximum</u> resource limits
$p_i$	Priority of task i
$f_{jw}$	Substitution efficiency of resource j for work requirement w ( $>0$ )
$s_{iw}$	Sequence of work requirement w in task i ( $>0$ )
$e_{iwj}$	Efficiency of resource j used for work w on task i

##### Decision Variables

$y_{iwj}$	Allocation of resource j to task i resulting in work w
-----------	--

##### Formulation

$$\begin{aligned}
 & \text{MAX } \sum_{st} \sum_{iwj} e_{iwj} y_{iwj} \\
 & \sum_j y_{iwj} \stackrel{o}{=} (\underline{r}_{iw}, \bar{r}_{iw}) ; (\underline{q}_{iw}, \bar{q}_{iw}) \text{ for all } i, w \quad (1) \text{ (GUB)} \\
 & \sum_{iw} e_{iwj} y_{iwj} \stackrel{o}{=} (\underline{a}_{jk}, \bar{a}_{jk}) ; (\underline{z}_{jk}, \bar{z}_{jk}) \text{ for all } j \quad (2) \\
 & y_{iwj} \geq 0 \quad \text{for all } i, w, j \quad (3) \quad (GN_k)
 \end{aligned}$$



$(GN_k)$  uses the notation of (IP). However, the dimensions of  $(GN_k)$  discriminate between resources consumed,  $j$ , and work completed,  $w$ , explicitly representing substitutability of resources. Constraints (1) encourage allocation of sufficient work resources, while constraints (2) indicate the desired mix of employable unit resources. Constraints (3) require non-negative resource allocations.  $(GN_k)$  is an elastic generalized network (Brown and McBride 1984).

Efficiency of resource  $j$  used for work  $w$  on task  $i$  is defined

$$e_{ijw} \equiv e^{-\ln f_{jw} + (p_i - 1)/10 + s_{iw}^+/10 + \bar{d}_{ik}/\sigma_k}, \quad (4)$$

where  $s_{iw}^+ = \max \{0, s_{iw} - t\}$ ,  $t$  is the last time period of this allocation, and  $\sigma_k$  is the unit speed of advance. The efficiency (4) employs the readiness and substitutability of resources via  $f_{jw}$ .  $s_{iw}^+$  reduces efficiency if the work  $w$  should not be started until period  $s_{iw}$ .

If model (IP) has been used for operational assignment,

$$\bar{d}_{ik} \equiv \max \{0, d_{ik} - \sigma_k/2\}. \quad (5)$$

If unit  $k$  is to be advanced toward, or to, location  $l$  by model  $(IP_L)$ ,

$$\bar{d}_{ik} \equiv \max \{0, d_{lk} - \sigma_k/2\} + g_{il}. \quad (6)$$

These distance costs  $\bar{d}_{ik}$  in (5) or (6), and penalties  $q_{iw}$ ,  $\bar{q}_{iw}$ ,  $z_{jk}$ , and  $\bar{z}_{jk}$  are all intended to yield the same objective function units. For our tests,  $q_{iw} \equiv 100/p_i s_{iw}^+$ , and  $\bar{q}_{iw} \equiv 100$ .

## V. CONSIDERATION OF LOGISTICS

The efficiency with which a unit completes a task depends heavily upon logistical considerations. If a unit is remote from a task, or must be moved, its efficiency suffers. Figure 1 shows an idealized situation with unit  $k$ , task  $i$ , and location  $l$ .

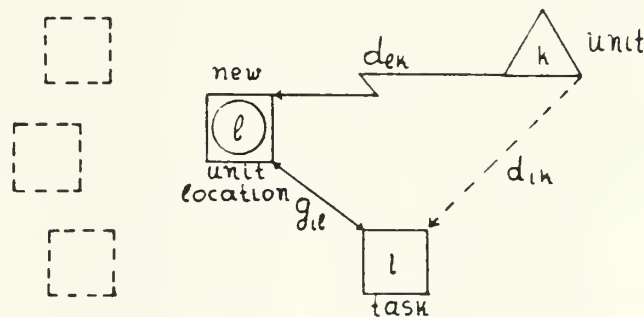


Figure 1: Idealized Geographic Logistical Scenario

Model (IP) assigns tasks to units relying exclusively upon  $d_{ik}$ . ( $IP_L$ ) moves units to new locations and assigns tasks to be performed from these new unit locations. ( $IP_L$ ) recognizes  $d_{lk}$  and  $g_{il}$ . The distances  $d_{ik}$  and  $d_{lk}$  are surrogates for logistical costs of assignment during the ensuing time period. Clearly, ( $IP_L$ ) is more appropriate for situations in which unit movements are expected, (IP) when they are not. ( $IP_L$ ) provides the decision maker with a better opening gambit than does (IP) if the scenario involves significant initial redeployment of units.

Tactical allocation models (GN) are given unit and task assignments and planned unit movements. Therefore, (GN) can allocate resources using

any logistic efficiency function of assigned distances, and of other attributes induced only from assignment such as weather effects, speed of unit movement, etc. (GN) can also substitute resources at somewhat reduced efficiency as well as prioritizing their immediate application. Given a fairly reasonable operational assignment, (GN) provides a high-resolution work plan with rich logistic detail and good face validity.



## VI. AN EXAMPLE SCENARIO

We demonstrate ARES with an example constructed for Engineering Battalions of the Hellenic Army. The mission scenario involves 20 tasks repairing major damage to public works following an earthquake. For our purposes, there are 14 units, each endowed with some of 25 resources. Figure 2 shows the units and tasks from the ARES input script. In the United States, the Department of the Army defines unit types in (1976), and task standards in (1973a).

UNIT LABELS, LOCATIONS, AND PRIOR ASSIGNMENTS						(IPL)	
UU	ILABEL	I	IX-COORD	IY-COORD	ISOA	LL	PP
1	1ST COMBAT BN	5-35	20.30	20.30	150		
2	2ND COMBAT BN	5-35	04.50	05.50	150		
3	3RD COMBAT BN	5-155	07.00	17.25	150		
4	4TH COMBAT BN	5-155	04.90	08.70	150		
5	1ST CONSTR BN	5-115	05.20	14.50	100		
6	2ND CONSTR BN	5-115	13.30	18.75	100		
7	3RD CONSTR BN	5-115	11.05	15.60	100		
8	4TH CONSTR BN	5-115	08.50	15.25	100		
9	1ST AIRBCR BN	5-195	13.00	05.00	200		
10	2ND AIRBCR BN	5-195	20.30	20.30	200		
11	1ST HIGH.EQUI.CO	5-5B	00.40	12.85	150		
12	1ST ENG ARM BT	5-145	13.20	08.60	120		
13	2ND ENG ARM BT	5-145	09.25	02.90	120		
14	3RD ENG ARM BT	5-145	08.50	15.25	120		

TASK LABELS, LOCATIONS, PRIORITIES, AND PRIOR ASSIGNMENTS						(IP)(IPL)	
TT	ILABEL	I	IX-COORD	IY-COORD	IPRI	UU	LL PP
1	ADMIN BUILDING	AA1051	09.55	05.90	1		
2	ADMIN BUILDING	AA1051	09.60	11.60	1		
3	ADMIN BUILDING	AA1101	01.90	06.95	1		
4	HOSPITAL 100 BED	GH0111	10.75	07.45	1		
5	HOSPITAL 200 BED	GH0211	09.60	11.60	1		
6	HOSPITAL 100 BED	GH0131	09.55	05.90	1		
7	HOSPITAL 100 BED	GH0131	01.90	06.95	1		
8	RAILROAD BRIDGE	B61643	09.70	05.90	1		
9	RAILROAD BRIDGE	861512	07.90	09.45	1		
10	ROAD BRIDGE 50'	854101	10.70	07.20	2		
11	ROAD BRIDGE 100'	854109	09.70	05.89	1		
12	ROAD BRIDGE 70'	854104	08.30	09.30	2		
13	ROAD 3.5 MILES	853120	10.80	06.15	2		
14	ROAD 4.7 MILES	853122	10.76	07.50	2		
15	ROAD 5.5 MILES	853128	09.60	06.30	2		
16	ROAD 6.8 MILES	853124	02.35	07.25	2		
17	ROAD 6.0 MILES	853120	10.80	06.95	2		
18	WATER TANK-DIST-SUP	NO1	10.85	06.90	1		
19	WATER TANK-DIST-SUP	NO2	09.65	06.35	1		
20	WATER TANK-DIST-SUP	NO3	09.10	10.85	1		

Figure 2: Units and Tasks of Example

The georeference coordinates of units and tasks are given in Figure 1 for the situation depicted in Figures 3, 4 and 5.

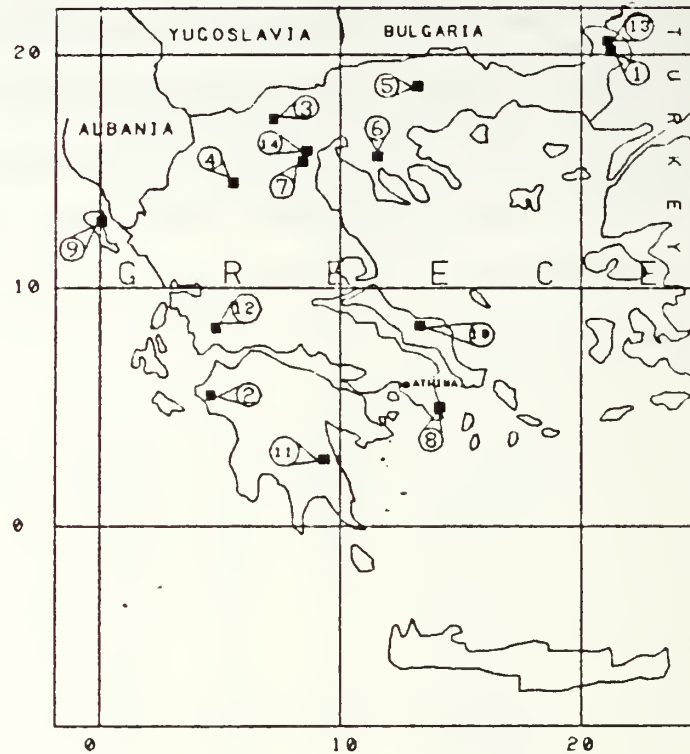


Figure 3: Initial Geographic Locations of Units. (Coordinates displayed are a georeference in common with the following figures.)

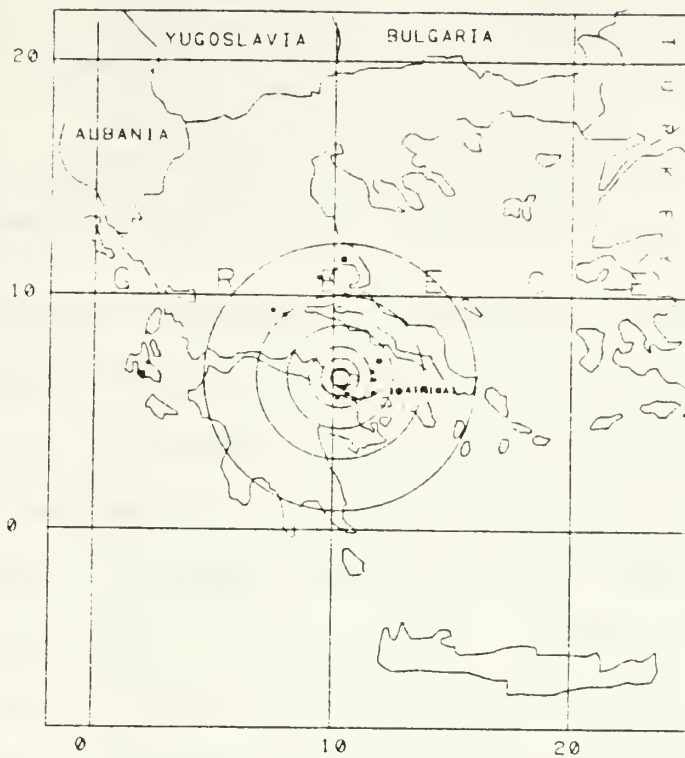


Figure 4: Earthquake Epicenter.

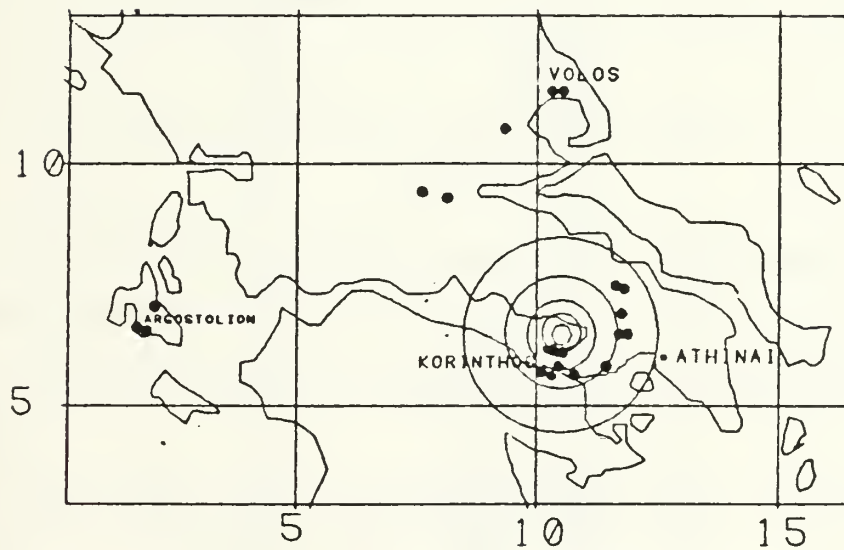


Figure 5: Geographic Locations of Tasks. Geographic locations of damaged public works and earthquake epicenter are shown.

A georeference system is used to generate coordinate-to-coordinate distance costs, which appear in the ARES input script.

The resource requirements for Task 1 ("ADMIN. BUILDING AA1051"), a disaster relief facility, are given in Figure 6 and the input script. Resource requirements such as these are available in standard engineering reference manuals for a wide variety of task types (for instance, see unclassified sources from the United States Department of the Army (1973a, 1973b, and 1973c). We envision a taxonomy of standardized task data icons from which a particular set of requirements can be very quickly extracted and assembled for a scenario. The size of our resource requirements data base is modest but the resulting accuracy and level of detail are quite good. Better yet, data mobilization from a menu of such icons can be completed in minutes.

RESOURCE LABELS AND TASK 1 REQUIREMENTS			
RR	ILABEL	I	MAN HOURS
1	ENGIN-PION-APREN-HLPER		6648
2	SURVEYOR		70
3	CARPENTER		7557
4	ELECTRICIAN		940
5	PLUMBER		1740
6	MASON		1600
7	STRUCTURE SPECIAL.		0
8	HEAT-VENTILAT SPECIAL.		200
9	WELDER		0
10	PIPELINE		0
11	CRANE-SHOVEL OPER.		200
12	LOADER OPER.		350
13	DOZER OPER.		300
14	COMPRESSOR OPER.		0
15	DUMP TRUCK OPER.		400
16	CONCRETE MACHINE OPER.		0
17	GRADER OPER.		329
18	CRUSHER OPER.		0
19	DITCH MACHINE OPER.		300
20	ASPHALT SPECIAL.		0
21	POWER ROLLER OPER.		0
22	WATER DISTRIBUT. OPER.		0
23	POWER BOAT OPER.		0
24	ROTARY TILLER OPER.		0
25	SCRAPER OPER.		0

Figure 6: Resource Requirements of Task 1

The resources employable by Unit 1 ("1ST COMBAT BN"), a combat engineering battalion, are given in Figure 7 and in the input script. These resource endowments are in line with those given by the United States Department of the Army (1973a) with conversion to man hours from (1971).

RESOURCE LABELS AND UNIT 1 AVAILABILITIES

RR	ILABEL	I	IMIN	IMAX	IMIN PEN	IMAX PEN
2	SURVEYOR		405	450	10	10
3	CARPENTER		1215	1350	10	10
4	ELECTRICIAN		203	225	10	10
5	PLUMBER		1418	1575	10	10
6	MASON		1215	1350	10	10
7	STRUCTURE SPECIAL.		0	0	10	10
8	HEAT-VENTILAT SPECIAL.		0	0	10	10
9	WELDER		187	208	10	10
10	PIPELINE		0	0	10	10
11	CRANE-SHOVEL OPER.		1215	1350	10	10
12	LOADER OPER.		6165	6850	10	10
13	DOZER OPER.		4050	4500	10	10
14	COMPRESSOR OPER.		1013	1125	10	10
15	DUMP TRUCK OPER.		10935	12150	10	10
16	CONCRETE MACHINE OPER.		203	225	10	10
17	GRADER OPER.		1620	1800	10	10
18	CRUSHER OPER.		0	0	10	10
19	DITCH MACHINE OPER.		0	0	10	10
20	ASPHALT SPECIAL.		0	0	10	10
21	POWER ROLLER OPER.		0	0	10	10
22	WATER DISTRIBUT. OPER.		0	0	10	10
23	POWER BOAT OPER.		0	0	10	10
24	ROTARY TILLER OPER.		0	0	10	10
25	SCRAPER OPER.		0	0	10	10

Figure 7: Resource Endowment of Unit 1.

The input script also includes for each task the sequence of resource requirements expressed as the first period when the resource is best applied, and for each resource its substitution efficiency for other resources.

The scenario data constitutes about 1,000 records. However, these records derive from the unit, task, and resource definitions which are modest in number.



## VII. DESIGN AND IMPLEMENTATION

ARES is intended to help the decision maker, not to replace him.

Figure 8 shows the functional structure of ARES. The design is biased toward interactive use with review and intervention options at each stage of operational assignment and tactical allocation.

INITIALIZE:	Define NEW_SCRIPT
NEXT_PERIOD:	Redefine NEW_SCRIPT as OLD_SCRIPT
OP_ASSIGN:	Select Model ( $IP_L$ ) or (IP) Read OLD_SCRIPT Generate and Solve ( $IP_L$ ) or (IP) Record task and unit assignments on ASSIGN_FILE
REVIEW_IP:	Option to review assignments in ASSIGN_FILE either stop, or edit OLD_SCRIPT and GOTO OP_ASSIGN, or edit OLD_SCRIPT and/or ASSIGN_FILE and continue
TAC_ALLOC:	Read OLD_SCRIPT and store as SCRIPT Read ASSIGN_FILE and update SCRIPT assignments
UNIT-K:	Select ( $GN_k$ ), Generate and Solve Update SCRIPT resource requirements for work completed For next unit k REPEAT UNIT-K
NEW_SCRIPT:	Update SCRIPT unit locations and distance costs Write SCRIPT as NEW_SCRIPT
REVIEW_PERIOD:	Option to review results either stop, or edit OLD_SCRIPT and/or ASSIGN_FILE and GOTO OP_ASSIGN or edit NEW_SCRIPT and GOTO NEXT_PERIOD

Figure 8: ARES Functional Specification

ARES is implemented in FORTRAN H(Extended) and executes on an IBM 3033 AP computer using the VM/CMS operating system. Input scripts are read from files which may be viewed and modified with a full-screen editor such as XEDIT. (Software copyrights IBM Corporation.)

ARES uses the X-SYSTEM (Brown and Graves 1975) to solve  $(IP_L)$ ,  $(IP)$  and  $(GN_k)$  in real time. For each problem instance, problem generators directly convert input script data into an internal representation, the solver is invoked, and the solution is provided to a report writing program. ARES consists of a set of open subroutines and is executed with whatever preview, review, or other external interference is deemed desirable.

We envision cyclic use and review at varying levels of detail as a mission progresses over time. Accordingly, input scripts include the beginning period and number of periods in the ensuing time interval, which intrinsically scales time-dependent input data to the desired level of aggregation. We have tested ARES manually and by replacing the decision maker with a simulation which performs "judgement review" of successive solutions over time. This permits totally automatic evaluation of complete mission scenarios, and avoids tedious manual effort in our research. (A single time interval may generate 15, or 20 thousand lines of solution detail at the scale of our example scenario.)

The update of unit coordinate locations and distance costs is a simple surrogate for a more realistic and complicated georeference and mobility system. ARES estimates the direction and speed of advance of each unit during the time interval and relocates the unit. Then the distance costs are adjusted. If operating areas are known sufficiently in advance to

permit preparation of detailed georeference and mobility systems, ARES can accomodate the increased level of detail in real time (e.g., Brown, Ellis, Graves, and Ronen 1987). The update can also be used to degrade, or to amplify unit resource endowments and effectiveness to modify task resource requirements, or to change any other data artifact, providing a rich modelling arena.

## VIII. SCENARIO RESULTS

ARES has been used in simulation mode to completely plan mission scenarios from start to finish. For the earthquake scenario, Figure 9 shows the initial operational assignments of ( $IP_L$ ).

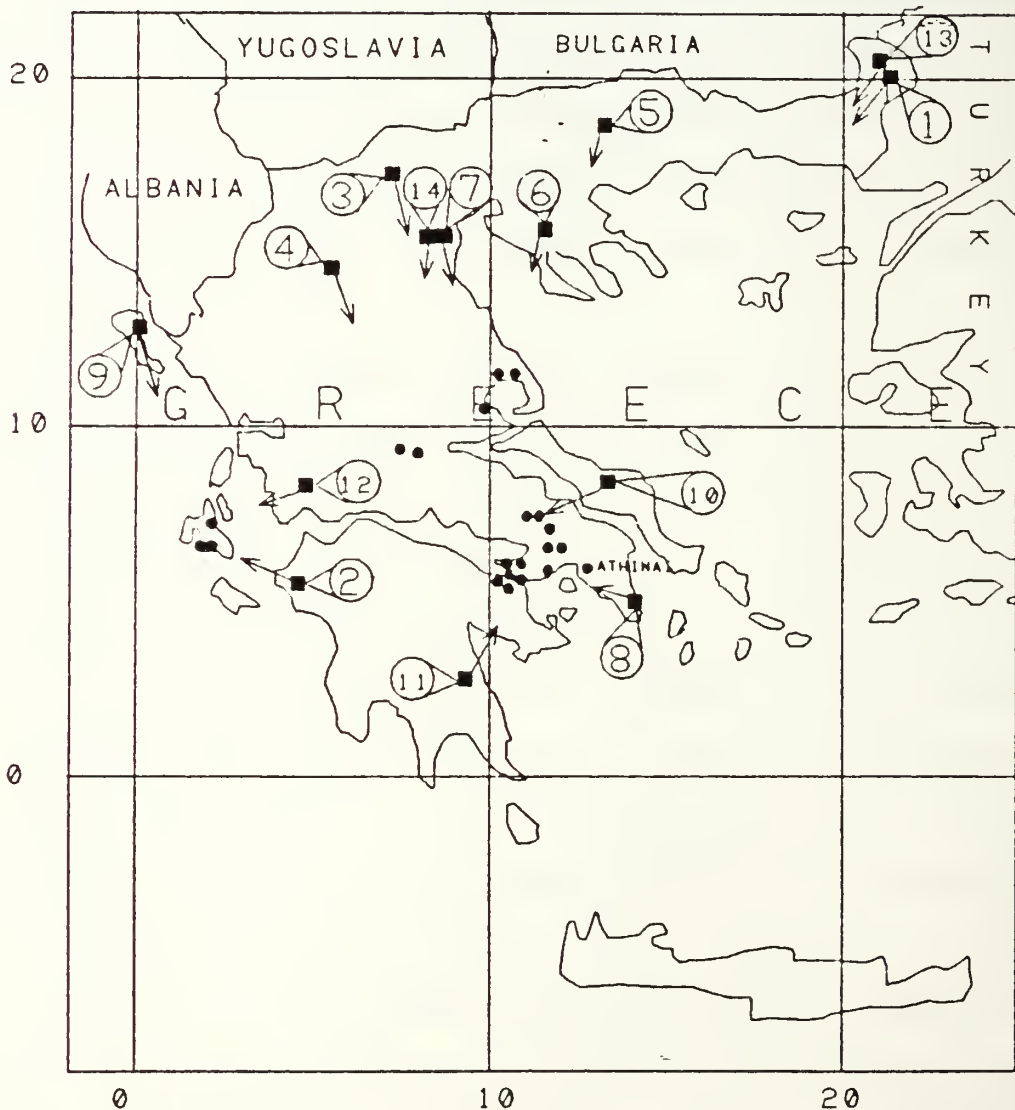


Figure 9: Initial Operational Assignments of Units. Directional vectors show the straight-line path and relative speed of advance  $\sigma_k$ .





The application of available resources, with allowable substitutions, is shown in Figure 11 for the 7 single-period time intervals required to complete the earthquake scenario.

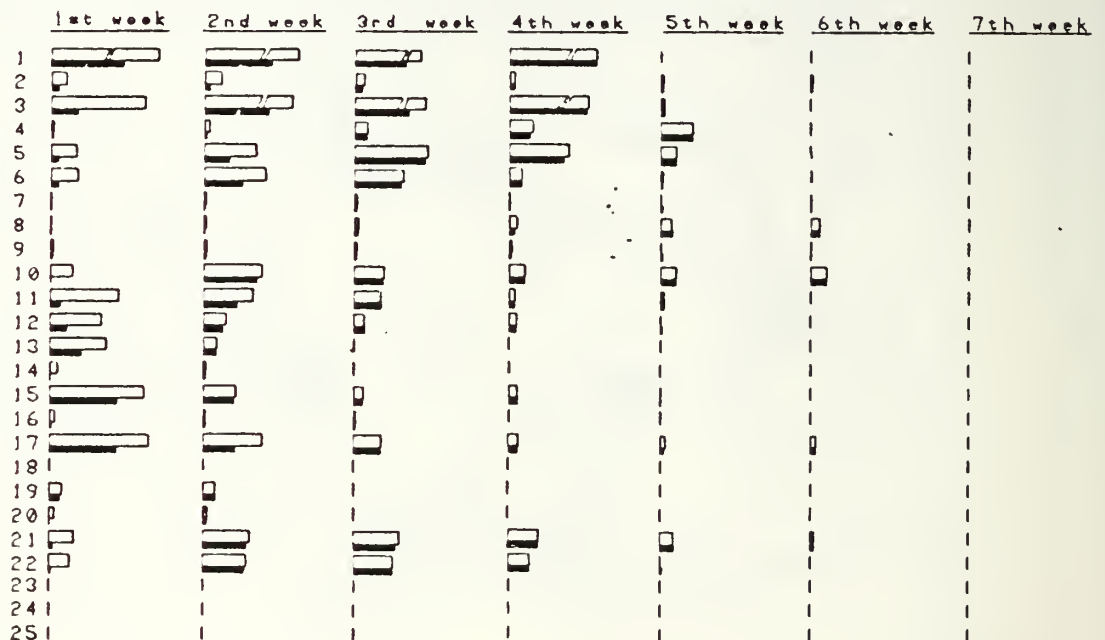


Figure 11: Resource Requirements and Work Completed. Each row represents a resource requirement over time-interval columns. The white bars depict resource requirements by time interval; the black bars show the relative fulfillment of the requirements. Broken bars are out of scale. From each time interval to the next the requirements are reduced by the work completed and amplified by new sequence-dependent requirements. In this scenario, 7 weekly time intervals are required to complete all tasks.

## IX. COMPUTATIONAL EXPERIENCE

Extensive computational experience reveals that the operational assignment models (IP) and especially ( $IP_L$ ) are most difficult to solve at the beginning of a scenario, and get progressively easy in later time intervals. The size of these models varies with the number of mandated assignments, impossible assignments, and the non-zero density of resource availabilities and remaining requirements. (IP) typically has about 340 constraints, 268 binary variables, and 6,200 non-zero consumption coefficients. The linear program continuous relaxation can be generated and solved in about 5 seconds, and an optimal binary solution is achieved in another second, or so.

( $IP_L$ ) has about 1,000 constraints, 645 binary variables, and 8,000 rather unwieldy non-zero gross resource requirement and net resource availability coefficients.

The linear program continuous relaxation of ( $IP_L$ ) proved impossible to solve by direct assault. Prior work by Brown and Graves for Bausch (1982) on large-scale set partitioning problems and later refinements by Brown, Graves and Ronen (1987) suggested an alternate means of attack: a problem cascade.

Briefly, the rows of constraints and columns of variables are lexicographically sorted to place short rows first accompanied by other rows and columns with intersecting non-zero coefficients, and longer rows later with their own intersecting rows and columns.

The problem cascade proceeds by activating a set of constraints, relaxing all other constraints, and activating a set of variables, fixing

all other variables to their last-known values. This problem is solved, the new values of the active variables recorded, and another problem specified in the building problem cascade. The last problem in the cascade activates all constraints and variables (precisely the problem found intractable above) and solves it by starting with an advanced solution recorded from the last-known values of variables solving previous problems in the cascade.

( $IP_L$ ) resisted even the problem cascade until a new heuristic cascade strategy was adopted which activates the shortest 1/2 of constraints and their associated variables, then the shortest 3/4, then 7/8, and so forth until the last constraint is added and the problem is solved. Remarkably, this approach has been absolutely reliable and robust, while most others fail or prove unruly.

Generation and complete problem cascade solution of the continuous relaxation of ( $IP_L$ ) now requires about 10 seconds.

An acceptable binary solution to ( $IP_L$ ) is achieved in another second, or two.

We do not routinely seek optimal binary solutions to ( $IP_L$ ), which we refer to as "perfect misfits". The gross resource requirements and net resource availabilities in ( $IP_L$ ) are rough logistic estimates, calibrated by actual field experience but ultimately just approximate target performance levels. For interesting operational assignments (i.e., early in the scenario) there are simply no feasible solutions; the goal is to guess where to send units so that they can peremptorily cope with the mission at hand with maximal effectiveness. Accordingly, we accept in practice binary solutions which may be as much as 25% greater than an

optimal lower bound in total value, including constraint violation penalties. Experimentally, we have determined at additional computational cost (as much as 5 minutes per trial) that these binary solutions are actually almost always within a few percent of the true optimum.

A decision maker can help ARES with its operational assignments or completely specify a solution with manual assignment features. Our experience suggests that the decision maker can express some non-quantifiable guidance in this fashion, but can not hope to apply a remotely competitive global perspective. Manual competition with ARES reveals that model computation effort is amply justified by the quality of operational assignments achieved. The operational assignment models, especially  $(IP_L)$ , produce solutions no decision maker is likely to discover. Some of these solutions have yielded remarkable insights. The initial operational commitment of units is arduous and crucial to mission success.  $(IP_L)$  is worth the computational investment.

By contrast, the tactical allocation models  $(GN)$  are easy to solve even in the cases where heroic substitution of resources are required. The size of each  $(GN_k)$  varies with the number of tasks assigned to the unit, and the non-zero densities of resource availabilities, remaining requirements, and allowable substitutions. For our scenario, a typical instance of  $(GN_k)$  has about 70 constraints and 1,190 variables, and is generated and solved in less than 0.1 second. Stress tests with 525 constraints and 12,500 variables require less than a second.



## X. DISCUSSION AND CONCLUSION

The subtlety of operational assignment has surprised us, as has the ease of detailed tactical allocation. Operational assignments are delicate decisions, and the success of entire missions appears to be very sensitive to minute details---precisely the considerations a hard-pressed decision maker would likely overlook in haste.

Extensive mechanisms have been provided in ARES to encourage manual review and coercion of solutions. However, there have been very few cases in which such guidance improved solutions and many instances in which minor manual adjustments of operational assignments inflicted great disruption. For example, some operational assignments of ( $IP_L$ ) "cross-locate" units in the sense that a pair of units will each be collocated with a task assigned to the other. This superficial blemish can easily be masked by manual intervention or by automated solution editing. Surprisingly, the removal of cross-locations frequently increases the logistic cost of the solution: there is a very delicate balance of logistic support of task cohorts assigned to specialized units. Cross-location can actually make a great deal of sense in practice.

Manual intervention can work well in cases inviting human judgement. For instance, nearly completed tasks or tasks which have been in progress for long intervals can enjoy efficiencies not apparent to our models. The decision maker can easily declare tasks completed when minor requirements remain, or when it is clear that the models are unduly influenced by a minor requirement.

Operational assignments can be restricted so that units are not moved from their initial new locations until the work in their logistic influence has been completed. Surprisingly, this restriction is rarely needed in practice, and in those cases in which multiple relocations are indicated great efficiencies accrue to the mission as a whole. We view this insight as a strong validation of the modelling philosophy underlying ARES.

Fortuitous design decisions to separate operational assignment and tactical allocation models, to decompose time intervals, and to couple the resulting restricted components with simulation and human intervention options have yielded more than the intended benefits. Our original motives were to capture as much reality as possible while still rendering models capable of quick, responsive solution.

The decomposed design also naturally accomodates features which are otherwise difficult to provide. For instance, partial orderings within tasks can be introduced. Also, discussions with Professor Wayne Hughes have suggested the technical feasibility of campaign analysis, two-sided gaming, and force-on-force applications of ARES. In these contexts, the coupling with simulation enhances our capabilities enormously.

ARES was originally designed for use on an IBM-PC. There is no compelling reason not to use this microcomputer, but we encountered a few practical limitations. An arbitrary 640 kilobyte memory region limitation and crippling errors in the FORTRAN compilers available for the IBM-PC present artificial conversion costs which we are not willing to bear. When these unfortunate shortcomings of the IBM-PC are repaired, conversion might be reconsidered. Calibration tests project solution times on the

order of 2 minutes per time interval on IBM-PC/AT with a math co-processor and internal clock speed of 8 megahertz.

## LIST OF REFERENCES

- Bausch, D. O., "Computational Advances in the Solution of Large-Scale Set Covering and Set Partitioning Problems," M.S. Thesis, Naval Postgraduate School (October 1982) 77 pages.
- Bracken, J., Falk, J. E., and Karr, A. F., "Two Models for Optimal Allocation of Aircraft Sorties," Operations Research, 23, (1975), 979-995.
- Bracken J., and McGill, J. T., "Defense Applications of Mathematical Programs with Optimization Problems in the Constraints," Operations Research, 22, (1974), 1086-1101.
- Brown, G. G., Ellis, C. E., Graves, G. W., and Ronen D., "Real-Time Wide Area Dispatch of Mobil's Petroleum Tank Trucks," Interfaces (to appear 1987).
- Brown, G. G. and Graves, G. W., "Design and Implementation of a Large Scale (Mixed Integer, Nonlinear) Optimization System," paper presented at ORSA/TIMS, Las Vegas, November 1975.
- Brown, G. G., Graves, G. W. and Ronen, D., "Scheduling Ocean Transportation of Crude Oil," Management Science (to appear 1987).
- Brown, G. G. and McBride, R. M., "Solving Generalized Networks," Management Science, 30, (1984), 1497-1523.
- Chaiken, J. M., and Larson, R. C., "Methods for Allocating Urban Emergency Units: A Survey," Management Science, 19, (1972), P110-P130.
- Dantzig, G. B. and Van Slyke, R. M., "Generalized Upper Bounding Techniques," J. Computing and System Science, 1, (1967), 213-226.
- Department of the Army, Field Manual FM 5-35: Engineers' Reference and Logistic Data, Washington, D.C. (April 1971).
- Department of the Army, Technical Manual TM 5-301-1: Army Facilities Components System - Planning (Temperate), Washington, D.C. (September 1973).
- Department of the Army, Technical Manual TM 5-302: Army Facilities Components System Designs, Washington, D.C. (September 1973).
- Department of the Army, Technical Manual TM 5-303: Army Facilities, Components, System-Logistic-Data and Bills of Materials, Washington, D.C. (September 1973).
- Department of the Army, Field Manual FM 101-10-1: Staff Officers'

Field Manual Organizational, Technical, and Logistical Data,  
Washington, D.C. (July 1976).

- Fitzsimmons, J. A., "A Methodology for Emergency Ambulance Deployment," Management Science, 19, (1973), 627-636.
- Geisler, M. A., "The Simulation of a Large-Scale Military Activity," Management Science, 5, (1959), 359-368.
- Kaplan, S., "Readiness and the Optimal Redeployment of Resources," Naval Research Logistics Quarterly, 20, (1973), 625-638.
- Karchere, A., and Hoeber, F. P., "Combat Problems, Weapon Systems, and the Theory of Allocation," Operations Research, 1, (1953), 286-302.
- Kolesar, P, and Walker, W. E., "An Algorithm for the Dynamic Relocation of Fire Companies," Operations Research, 22, (1974), 249-274.
- Koopmans, T. C. (ed.), Activity Analysis of Production and Allocation, Wiley, New York, (1951).
- Swoveland, C., Uyeno, D., Vertinsky, I., and Vickson, R., "Ambulance Location: A Probabilistic Enumeration Approach," Management Science, 20, (1973), 686-698.



# INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22304-6145	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3. Department Chairman, Code 54 Department of Administrative Science Naval Postgraduate School Monterey, California 93943	1
4. Professor Gerald G. Brown, Code 55BW Department of Operations Research Naval Postgraduate School Monterey, California 93943	5
5. Professor Benjamin Roberts, Code 54RO Department of Administrative Science Naval Postgraduate School Monterey, California 93943	1
6. Hellenic Armed Forces Headquarters National Defense General Staff Department of Studies Holargos, Athens, GREECE	1
7. Hellenic Army Headquarters Corps of Engineers Department Holargos, Athens, GREECE	2
8. Hellenic Army Headquarters Defense Plans and Programs Department Holargos, Athens, GREECE	1
9. Hellenic Army Headquarters Department of Studies Holargos, Athens, GREECE	1
10. Major Antonios L. Vassiliou - 7th Parodos Nicopoleos 7 Ioannina, GREECE	10

108  
18070 2  
12





DUDLEY, JOHN T. 1934-1984  
NATIONAL DEFENSE UNIVERSITY SCHOOL  
MONTHERLY, CALIFORNIA 95045-6002

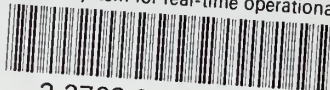
Thesis  
V3421 Vassiliou  
c.1 ARES a system for real-  
time operational and tac-  
tical decision support.

Thesis  
V3421 Vassiliou  
c.1 ARES a system for real-  
time operational and tac-  
tical decision support.



thesV3421

ARES a system for real-time operational



3 2768 000 71056 0

DUDLEY KNOX LIBRARY